An a posteriori verification method for generalized real-symmetric eigenvalue problems in large-scale electronic state calculations

Takeo Hoshi[a], Takeshi Ogita[b], Katsuhisa Ozaki[c], Takeshi Terao[d]

[a]*Department of Applied Mathematics and Physics, Tottori University, Japan*
[b]*Division of Mathematical Sciences, Tokyo Woman's Christian University, Japan*
[c]*Department of Mathematical Sciences, Shibaura Institute of Technology, Japan*
[d]*Graduate School of Engineering and Science, Shibaura Institute of Technology, Japan*

Abstract

An a posteriori verification method is proposed for the generalized real-symmetric eigenvalue problem and is applied to densely clustered eigenvalue problems in large-scale electronic state calculations. The proposed method is realized by a two-stage process in which the approximate solution is computed by existing numerical libraries and is then verified in a moderate computational time. The procedure returns intervals containing one exact eigenvalue in each interval. Test calculations were carried out for organic device materials, and the verification method confirms that all exact eigenvalues are well separated in the obtained intervals. This verification method will be integrated into EigenKernel (`https://github.com/eigenkernel/`), which is middleware for various parallel solvers for the generalized eigenvalue problem. Such an a posteriori verification method will be important in future computational science.

Keywords: verification method, generalized real-symmetric eigenvalue problem, electronic state calculation, supercomputer,
2000 MSC: 65F15, 65G20

1. Introduction

A crucial issue in verification methods is application to large-scale scientific or industrial computations on supercomputers. Many numerical solvers have been proposed for modern massively parallel supercomputers, and application researchers would like to compare solvers both in terms of computational speed and reliability. The concept of a posteriori verification methods is proposed in order to meet the needs of application researchers.

A posteriori verification methods have the workflow shown in Fig. 1. An approximate solution is first obtained and then verified. The former and latter procedures are referred to as a solver and a verifier, respectively. The goal of the present study is to integrate the verifier routine, as an optional function, to existing numerical solver libraries.

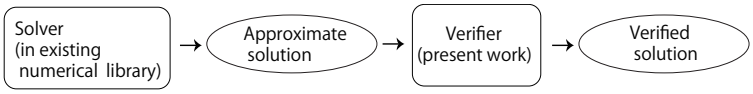

Figure 1: Schematic diagram of the workflow with an a posteriori verification method.

The present research is motivated by large-scale electronic state calculation, a major field in computational material science and engineering. As explained in Appendix A, a mathematical model is used for the fundamental Schrödinger-type equation, and the problem is reduced to the generalized real-symmetric matrix eigenvalue problem

$$Ax_k = \lambda_k Bx_k \tag{1}$$

under the generalized orthogonality condition

$$x_i^{\mathrm{T}} B x_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where both A and B are real-symmetric $n \times n$ matrices, with B being positive definite. Here, we assume that

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Applying our results to problems with complex Hermitian matrices is straightforward.

For large-scale electronic state calculation, many eigenvalues are densely clustered or almost degenerate, and distinguishing them numerically may be difficult. In order to obtain reliable results, we consider verification methods for generalized eigenvalue problems. For the sake of completeness as verification methods, we also need to take into account all numerical errors that occur when matrices A and B are generated from the fundamental Schrödinger-type equation. Although we do not consider the fundamental Schrödinger-type equation in detail herein, we briefly discuss this equation in Appendix A.

One of the authors (T. H.) developed a middleware EigenKernel [1, 2, 3] with various parallel solvers for generalized eigenvalue problems and plans to add a verifier routine. The total elapsed time T_{tot} is the sum of the times for solver T_{sol} and verifier T_{veri} ($T_{\mathrm{tot}} = T_{\mathrm{sol}} + T_{\mathrm{veri}}$). We attempt to construct the verifier algorithm so that the time for the verifier gives a moderate fraction ($T_{\mathrm{veri}} \leq T_{\mathrm{sol}}$). Since the verifier can use the highly optimized routines of matrix multiplication, the verifier is suitable for high-performance computing on supercomputers.

In the solver procedure, approximate solutions $(\hat{\lambda}_k, \hat{x}_k)$, $k = 1, 2, \ldots, n$, such that

$$A\hat{x}_k \approx \hat{\lambda}_k B\hat{x}_k \tag{3}$$

are obtained by any numerical solver algorithm. A verifier procedure gives the difference between the exact and approximate solutions, such as $|\lambda_k - \hat{\lambda}_k|$ or $\|x_k - \hat{x}_k\|$. If the relation $|\lambda_k - \hat{\lambda}_k| \leq r_k$ is obtained with a given positive number r_k, for example, this indicates that the exact solution (λ_k) lies in a disk having a center and radius of $\hat{\lambda}_k$ and r_k, respectively. For this purpose, a number of enclosure methods have been developed, e.g., [4, 5, 6]. In the present paper, we propose a method of enclosing all eigenvalues that is straightforward, efficient, and easy to implement on supercomputers. The proposed method is based on Yamamoto's theorem [7] and is essentially the same as the method proposed in a previous paper [6]. In other words, we specialize the previous method [6] for generalized real-symmetric eigenvalue problems. Note that it is not possible in general to state that a method is better or worse than other methods because this depends on the purpose. We compare the advantages and disadvantages of these enclosure methods in Section 3.

The a posteriori verification strategy is important mainly with regards to three aspects. First, numerical methods for the densely clustered eigenvalue problem have potential difficulties in computing reliable numerical solutions, as explained above. Second, various numerical algorithms have been proposed for efficient parallel computations that are suitable for current and next-generation supercomputers. Application researchers would like to compare these methods with respect to both computational speed and numerical reliability. Third, the emergence of machine learning has enhanced the design of computer architecture for the acceleration of low-precision (single- or half-precision) calculation. The efficient use of low-precision calculation, typically in mixed-precision calculation, will be important in any high-performance computational science field [8, 9]. A posteriori verification methods guarantee satisfactory numerical reliability when low-precision calculation is used.

The remainder of the present paper is organized as follows. Section 2 explains the physical and mathematical backgrounds. The proposed verification method and numerical examples are presented in Sections 3 and 4, respectively. Section 5 presents a summary and an outlook for future research.

2. Background

2.1. Large-scale electronic state calculation and densely clustered eigenvalue problem

The present electronic state calculation is briefly introduced in Appendix A. The matrix size n is approximately proportional to the number of the atoms, molecules, or electrons in the material. An eigenvalue (λ_k) or an eigenvector (x_k) indicates the energy and the wavefunction, respectively, of an electron.

The present research is motivated, in particular, by a previous study [10], in which we focused on the participation ratio [11, 12] defined for a vector $v \equiv (v_1, v_2, ..., v_n)$, as

$$P \equiv P(v) = \left(\sum_{j=1}^{n} |v_j|^4 \right)^{-1}. \tag{4}$$

The participation ratio is a measure of the spatial extension of the electronic wavefunction and governs the electronic device properties. A dense eigenvector, *i.e.*, a vector that has only a few components that are negligible in terms of absolute value, has a large participation ratio. The corresponding electronic wavefunction is extended through the material and can contribute to electrical current. A sparse eigenvector, *i.e.*, a vector that has only a few components that are large in terms of absolute value, indicates a small participation ratio. The corresponding electronic wavefunction is localized in the material and cannot contribute to electrical current. An interesting research target in a large-scale problem is an 'intermediate' electronic wavefunction or a wavefunction that shows intermediate properties between extended and localized wavefunctions. Such 'intermediate' wavefunctions appear, for example, in Fig. 1 of Ref. [12] or Fig. 3 of Ref. [10].

The densely clustered eigenvalue problem in (1) appears among large-scale calculations and is illustrated in Fig. 2. In the problem, the difference of sequential eigenvalues $\delta_k \equiv \lambda_{k+1} - \lambda_k$, $k = 1, 2, \ldots, n-1$, tends to be proportional to $1/n$ ($\delta_k \propto 1/n$). Consequently, many eigenvalues are densely clustered or almost degenerate ($\delta_k \to 0$) in a large-matrix problem ($n \to \infty$) and distinguishing these eigenvalues numerically may be difficult.

It is crucial to distinguish each eigenvalue numerically among densely clustered eigenvalues, because the participation ratio and other physical quantities are defined for each eigenvector. If two calculated eigenvalues $\hat{\lambda}_k$ and $\hat{\lambda}_{k+1}$ cannot be distinguished in the numerical calculation, or if the two eigenvalues are recognized, unphysically, to be degenerate, then the corresponding eigenvectors $\hat{x}_k$ and $\hat{x}_{k+1}$ cannot be defined uniquely. In this case, the participation ratio values $P(\hat{x}_k)$ and $P(\hat{x}_{k+1})$ are not defined uniquely and any discussion of these values will be meaningless.

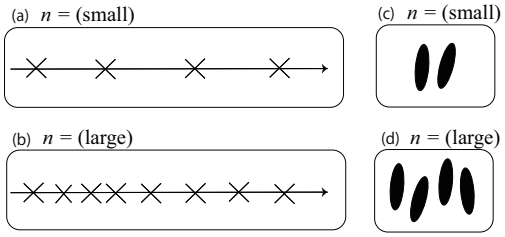

Figure 2: Schematic diagram of a densely clustered eigenvalue problem in large-scale electronic state calculations. (a), (b) Eigenvalue distribution in (1) with (a) small or (b) large matrix size n. A cross indicates an eigenvalue on the real axis. The difference of sequential eigenvalues tends to be proportional to $1/n$. (c), (d) Materials with (c) small or (d) matrix size n. Ovals indicate molecules. The matrix size n is proportional to the size of the molecules n_{mol} ($n \propto n_{\mathrm{mol}}$).

2.2. Numerical solvers for the generalized eigenvalue problem

Here, an overview is given for the parallel dense-matrix solver of the generalized eigenvalue problem of (1), in particular, for the variety of used algorithms. The solver algorithm for (1) consists of four procedures: (i) Cholesky decomposition of B

$$B = R^{\top} R, \tag{5}$$

with an upper triangular matrix R, (ii) reduction to the standard eigenvalue problem (SEP)

$$A' y_k = \lambda_k y_k, \tag{6}$$

with

$$A' \equiv R^{-\top} A R^{-1}, \tag{7}$$

(iii) solution of the standard eigenvalue problem (6), and (iv) transformation of the eigenvectors

$$x_k = R^{-1} y_k. \tag{8}$$

The set of procedures (i), (ii), and (iv) is referred to as reducer, and procedure (iii) is referred to as the SEP solver.

Although ScaLAPACK [13, 14] is the *de facto* standard parallel numerical library, this library was developed mainly in the 1990s, and several routines exhibit severe bottlenecks on modern massively parallel supercomputers. Novel solver libraries of ELPA [15, 16] and EigenExa [17, 18] were proposed in order to overcome the bottlenecks. The ELPA code was developed in Europe under tight collaboration between computer scientists and material science researchers, and its main target application is FHI-aims [19, 20], which is a well-known electronic state calculation code. The EigenExa code, on the other hand, was developed at RIKEN in Japan. Importantly, the ELPA code has routines optimized for x86, IBM Blue-Gene, and AMD architectures, whereas the EigenExa code was developed to be optimal mainly on the K computer, which is a Japanese flagship supercomputer. Both ScaLAPACK and ELPA provide the reducer routines, and all of ScaLAPACK, ELPA, and EigenExa provide the SEP solver routines.

Since the computational performance depends both on the problem and the architecture, it is, in principle, possible to construct a 'hybrid' workflow in which the reducer routine is chosen from one library and the SEP solver routine is chosen from another library, so as to realize optimal performance. The middleware EigenKernel was developed in order to realize such hybrid workflows. An obstacle to realizing the hybrid workflow is the difference of matrix distribution schemes between different libraries. EigenKernel provides data conversion routines between libraries and surmounts this obstacle.

Figure 3 shows the possible workflows for a future version of EigenKernel with the a posteriori verification routine. The SEP solvers and the reducers in Fig. 3 are briefly explained. The SEP solver and the two reducers in ScaLAPACK are the traditional routines. The SEP solvers of 'ELPA1' and 'Eigen_s' are also based on the traditional algorithm with tridiagonalization. The other two solvers 'ELPA2' and 'Eigen_sx' and the reducer in ELPA are based on non-traditional algorithms for better performance in massive parallelism. The detailed algorithms for these routines are found in Ref. [3].

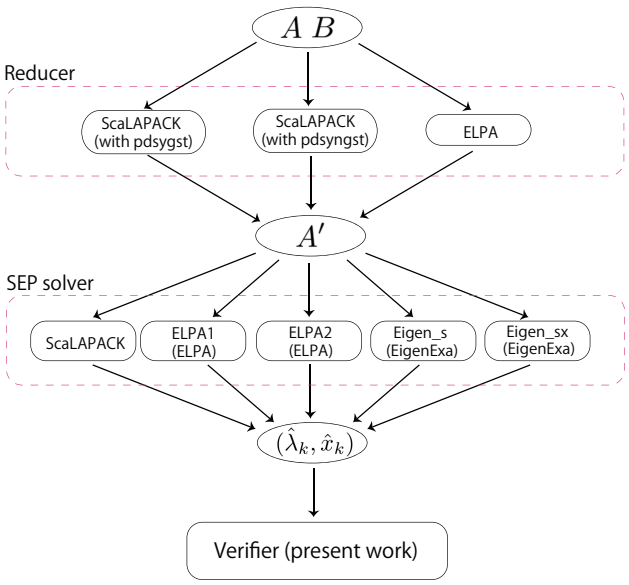

Figure 3: Schematic diagram of the possible hybrid workflows for a future version of EigenKernel with the a posteriori verification routine. Two routines in ScaLAPACK and one routine in ELPA are available for the reducer, whereas one routine in ScaLAPACK, two routines in ELPA, and two routines in EigenExa are available for the SEP solver. The a posteriori verification routine is commonly used among the workflows.

2.3. Verified numerical computations

We briefly explain how to obtain mathematically rigorous numerical results using floating-point arithmetic. Let $\mathbb{F}$ and $\mathbb{IF}$ be sets of floating-point numbers and intervals, respectively. We use bold-faced letters for interval matrices, the elements of which are intervals. For an interval matrix $\mathbf{C}$, $C_{\inf}$ and $C_{\sup}$ denote the left and right endpoints, respectively, such that $\mathbf{C} = [C_{\inf}, C_{\sup}]$, i.e., $\mathbf{C}_{ij} = [(C_{\inf})_{ij}, (C_{\sup})_{ij}]$ for all (i, j) pairs, which is known as "inf-sup" form. In addition, C_{mid} and C_{rad} denote the midpoint and the radius of $\mathbf{C}$, respectively, such that $\mathbf{C} = [C_{\mathrm{mid}} - C_{\mathrm{rad}}, C_{\mathrm{mid}} + C_{\mathrm{rad}}]$, which is known as "mid-rad" form. Let $fl(\cdot)$, $fl_{\triangledown}(\cdot)$, and $fl_{\triangle}(\cdot)$ be computed results by floating-point arithmetic as defined in IEEE 754 with rounding to the nearest (roundTiesToEven), rounding downwards (roundTowardNegative), and rounding upwards (roundTowardPositive), respectively. For a given matrix $C = (c_{ij}) \in \mathbb{R}^{n\times n}$, the notation $|C|$ indicates $|C| = (|c_{ij}|) \in \mathbb{R}^{n\times n}$, and the same applies to vectors, i.e., the absolute value is taken componentwise.

Next, we review basic interval matrix multiplication (cf. [21]). For two point matrices $P, Q \in \mathbb{F}^{n\times n}$, the matrix product $PQ \in \mathbb{R}^{n\times n}$ can be enclosed as

$$PQ \in [fl_{\triangledown}(PQ),\ fl_{\triangle}(PQ)], \tag{9}$$

where two matrix multiplications are required. For a point matrix $P \in \mathbb{F}^{n\times n}$ and an interval matrix $\mathbf{Q} \in \mathbb{IF}^{n\times n}$, the product $P\mathbf{Q}$ can efficiently be enclosed using mid-rad form of $\mathbf{Q}$ as

$$P\mathbf{Q} \subset [fl_{\triangledown}(PQ_{\mathrm{mid}} - T),\ fl_{\triangle}(PQ_{\mathrm{mid}} + T)], \quad T = fl_{\triangle}(|P|Q_{\mathrm{rad}}), \tag{10}$$

which involves three matrix multiplications. Although the inf-sup form can also be used for calculating the enclosure of $P\mathbf{Q}$, the inf-sup form cannot be written with products of point matrices simply, so that it is much more difficult for the inf-sup form to achieve high-performance in practice, as compared to the mid-rad form [21]. If $\mathbf{Q}$ is given by the inf-sup form $[Q_{\inf}, Q_{\sup}]$, we can easily transform $\mathbf{Q}$ into the mid-rad form, for example, by

$$Q_{\mathrm{mid}} = fl_{\triangle}((Q_{\inf} + Q_{\sup})/2), \quad Q_{\mathrm{rad}} = fl_{\triangle}(Q_{\mathrm{mid}} - Q_{\inf}),$$

which satisfies $[Q_{\inf}, Q_{\sup}] \subset [Q_{\mathrm{mid}} - Q_{\mathrm{rad}}, Q_{\mathrm{mid}} + Q_{\mathrm{rad}}]$.

There exist several implementations of the above interval arithmetic for matrix multiplication, e.g., C-XSC [22], a C++ library, and INTLAB [23], a Matlab/Octave toolbox for verified numerical computations. Both C-XSC and INTLAB share the common feature that they use Basic Linear Algebra Subprograms (BLAS) routines. In other words, we can efficiently implement interval matrix multiplication using PBLAS, the parallel version of BLAS, on distributed computing environments, as long as directed rounding in floating-point operations is available in BLAS routines for matrix multiplication and the reduction operation of summation.

3. A posteriori verification methods

3.1. Possible verification methods

Possible verification methods are discussed here. In order to measure the accuracy of the computed solution $(\hat{\lambda}_k, \hat{x}_k)$, application researchers often compute a norm of the residual vector, such as

$$\frac{\|A\hat{x}_k - \hat{\lambda}_k B\hat{x}_k\|_2}{\|\hat{x}_k\|_2}.$$

Although this quantity usually suffices to check whether the solver works correctly, it does not verify the accuracy of the computed eigenvalue. The following inequality is a known residual bound [5]:

$$\min_{1\le j\le n} |\lambda_j - \hat{\lambda}_k| \le \sqrt{\|B^{-1}\|_2}\frac{\|A\hat{x}_k - \hat{\lambda}_k B\hat{x}_k\|_2}{\sqrt{\hat{x}_k^{\top} B\hat{x}_k}}, \tag{11}$$

which is straightforwardly derived from Wilkinson's bound [24] for the standard eigenvalue problem. From the bound (11), we can confirm that some eigenvalue of (A, B) exists in the neighborhood of $\hat{\lambda}_k$ satisfying (11). However, we cannot determine whether $\hat{\lambda}_k$ is an approximation of the k-th eigenvalue of (A, B). In order to understand the electronic state of the problems correctly, it is crucial to determine the order of eigenvalues [25].

To our knowledge, we have the following two approaches to determine the order of eigenvalues of symmetric matrices:

(a) Compute all eigenpairs (pairs of eigenvalues and eigenvectors) and verify the error bounds of all computed eigenvalues (cf., e.g. [5, 6]).

(b) Compute an approximation $\hat{\lambda}_k$ of a target eigenvalue using Sylvester's law of inertia with $\mathrm{LDL}^\top$ decomposition [25], and verify that $\hat{\lambda}_k$ is an approximation of the k-th eigenvalue with an error bound (cf., e.g. [4]).

The advantages and disadvantages of each approach from a practical point of view are as follows:

- Approach (a) is simpler, numerically more stable, and easier to implement than Approach (b).
- Approach (a) can straightforwardly use highly optimized routines for matrix multiplication and eigenvalue decomposition.
- Approach (a) cannot exploit the sparsity of A and B, whereas Approach (b) can to a certain extent.

In the present paper, we adopt Approach (a) from the aspect of simplicity and efficiency of code development on supercomputers.

3.2. Proposed method

We attempt to obtain componentwise error bounds for computed eigenvalues $\hat{\lambda}_k$, $k = 1, 2, \ldots, n$. Let $X, D \in \mathbb{R}^{n\times n}$ denote a matrix comprising all generalized eigenvectors of (A, B) and a diagonal matrix of the corresponding generalized eigenvalues such that

$$X = [x_1, x_2, \ldots, x_n], \quad D = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

Let I denote the $n \times n$ identity matrix. Then, we have

$$\begin{cases} AX = BXD, \\ X^\top BX = I. \end{cases}$$

Let $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n] \in \mathbb{R}^{n\times n}$ and $\hat{D} = \mathrm{diag}(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_n) \in \mathbb{R}^{n\times n}$ be approximations of X and D, respectively. Suppose $\hat{X}$ is nonsingular. Then,

$$A\hat{X} \approx B\hat{X}\hat{D}, \quad \hat{X}^\top B\hat{X} \approx I \quad \Rightarrow \quad \hat{X}^{-1}B^{-1}A\hat{X} \approx \hat{D}, \quad (B\hat{X})^{-1} \approx \hat{X}^\top.$$

Since $\hat{X}^{-1}B^{-1}A\hat{X}$ is a similarity transformation of $B^{-1}A$, the eigenvalues of $\hat{X}^{-1}B^{-1}A\hat{X}$ are the same as those of $B^{-1}A$, and thus the generalized eigenvalues of (A, B).

Here, we attempt to compute an inclusion of $\hat{X}^{-1}B^{-1}A\hat{X}$. To this end, we introduce Yamamoto's theorem for verified solutions of linear systems. For given matrices $P = (p_{ij}), Q = (q_{ij}) \in \mathbb{R}^{n\times n}$, the notation $P \le Q$ indicates $p_{ij} \le q_{ij}$ for all (i, j), and the same applies to vectors, i.e., the inequality holds componentwise. Moreover, define $e \equiv (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$.

Theorem 1 (Yamamoto [7])**.** *Let A and C be real $n \times n$ matrices, and let b and $\hat{x}$ be real n-vectors. If $\|I - CA\|_\infty < 1$, then A is nonsingular, and*

$$|A^{-1}b - \hat{x}| \le |C(b - A\hat{x})| + \frac{\|C(b - A\hat{x})\|_\infty}{1 - \|I - CA\|_\infty}|I - CA|e.$$

In practice, we adopt an approximate inverse of A as C in Theorem 1.

In order to apply Yamamoto's theorem to componentwise error bounds for computed eigenvalues with the Gershgorin circle theorem, we present a variant of Yamamoto's theorem.

Theorem 2. *Let A, B, C, and $\hat{X}$ be real $n \times n$ matrices. If $\|I - CA\|_\infty < 1$, then A is nonsingular, and*

$$|A^{-1}B - \hat{X}|e \le |C(B - A\hat{X})|e + \frac{\|C(B - A\hat{X})\|_\infty}{1 - \|I - CA\|_\infty}|I - CA|e.$$

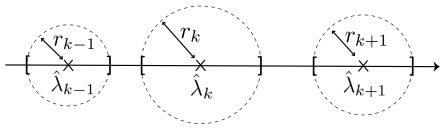

Figure 4: Schematic diagram of verified solution when all of the disks are separated, and each disk contains precisely one eigenvalue ($\lambda_k \in [\hat{\lambda}_k - r_k, \hat{\lambda}_k + r_k]$).

Proof. In a similar manner to the derivation of Yamamoto's theorem, and noting that for $P \in \mathbb{R}^{n\times n}$, $|P|e \le \|P\|_\infty e$, we have

$$\begin{aligned}
|A^{-1}B - \hat{X}|e &= |(CA)^{-1}C(B - A\hat{X})|e \\
&\le |(CA)^{-1}| \cdot |CR|e, \quad R \equiv B - A\hat{X} \\
&= |(I - (I - CA))^{-1}| \cdot |CR|e = |I + G + G^2 + \cdots| \cdot |CR|e, \quad G \equiv I - CA \\
&\le |CR|e + |G|(I + |G| + |G|^2 + \cdots)|CR|e \\
&\le |CR|e + \|CR\|_\infty |G|(I + |G| + |G|^2 + \cdots)e \\
&\le |CR|e + \frac{\|CR\|_\infty}{1 - \|G\|_\infty}|G|e,
\end{aligned}$$

which proves the theorem. □

We now consider a linear system $(B\hat{X})Y = A\hat{X}$ for Y. Then, we can regard $\hat{D}$ as its approximate solution and $\hat{X}^\top$ as an approximate inverse of $B\hat{X}$. Let Y, R, and G be defined as

$$\begin{cases} R \equiv \hat{X}^\top(A\hat{X} - B\hat{X}\hat{D}), \\ G \equiv \hat{X}^\top B\hat{X} - I. \end{cases} \tag{12}$$

If $\|G\|_\infty < 1$, applying Theorem 2 to the linear system $(B\hat{X})Y = A\hat{X}$ yields

$$|\hat{X}^{-1}(B^{-1}A)\hat{X} - \hat{D}|e = |(B\hat{X})^{-1}(A\hat{X}) - \hat{D}|e \le |R|e + \frac{\|R\|_\infty}{1 - \|G\|_\infty}|G|e \equiv r. \tag{13}$$

Recall that λ_i, $i = 1, 2, \ldots, n$, are the eigenvalues of $B^{-1}A$. For $\Lambda \equiv \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, the Gershgorin circle theorem implies

$$\Lambda \subseteq \bigcup_{k=1}^{n} [\hat{\lambda}_i - r_i, \hat{\lambda}_i + r_i]. \tag{14}$$

If all the disks $[\hat{\lambda}_i - r_i, \hat{\lambda}_i + r_i]$ are isolated, then all of the eigenvalues are separated, i.e., each disk contains precisely one eigenvalue of $B^{-1}A$ [26, pp. 71ff], as shown schematically in Fig. 4. If several disks are overlapped such that $|\hat{\lambda}_{k+1} - \hat{\lambda}_k| > r_k + r_{k+1}$ for some k, then some of the eigenvalues are degenerate or nearly degenerate. Moreover, if B is ill-conditioned, then the B-orthogonality of $\hat{X}$ may break down such that $\|G\|_\infty \ge 1$. In such a case, Theorem 2 cannot be applied, and the verification procedure must end in failure. Therefore, we need to check whether $\|G\|_\infty < 1$ in code development from the verification method.

In [6], a similar method has been proposed, which is essentially the same as the proposed method. The main difference between the method in [6] and the proposed method is that the former focuses on the non-symmetric case and is more general. On the other hand, the proposed method is specialized for the symmetric case, i.e., we can avoid complex arithmetic including the verification procedure and compute an approximate inverse of $B\hat{X}$ by utilizing $\hat{X}^\top \approx (B\hat{X})^{-1}$.

3.3. Code development

We explain how to obtain an upper bound of the vector r in (13) using only floating-point arithmetic. We first attempt to obtain upper bounds G' and R' of $|R| = |\hat{X}^\top(A\hat{X} - B\hat{X}\hat{D})|$ and $|G| = |\hat{X}^\top B\hat{X} - I|$ in (12) such that $|G| \le G'$ and $|R| \le R'$ as follows:

1. $\mathbf{C} \leftarrow B\hat{X}$ % Two matrix multiplications based on (9)
2. $\mathbf{F} \leftarrow \hat{X}^\top \mathbf{C}$ % Three matrix multiplications based on (10)
3. $\mathbf{W} \leftarrow \mathbf{F} - I$ % Negligible cost, $W_{\rm inf} \equiv fl_\triangledown(F_{\rm inf} - I),\ W_{\rm sup} \equiv fl_\triangle(F_{\rm sup} - I)$
4. $|G| \le \max(|W_{\rm inf}|, |W_{\rm sup}|) \equiv G'$
5. $\mathbf{F} \leftarrow A\hat{X}$ % Two matrix multiplications based on (9)
6. $\mathbf{C} \leftarrow \mathbf{C}\hat{D}$ % Negligible cost because $\hat{D}$ is a diagonal matrix
7. $\mathbf{C} \leftarrow \mathbf{F} - \mathbf{C}$ % Negligible cost, $C_{\rm inf}$ is overwritten by $fl_\triangledown(F_{\rm inf} - C_{\rm sup}),\ C_{\rm sup}$ is overwritten by $fl_\triangle(F_{\rm sup} - C_{\rm inf})$
8. $\mathbf{C} \leftarrow \hat{X}^\top \mathbf{C}$ % Three matrix multiplications based on (10)
9. $|R| \le \max(|C_{\rm inf}|, |C_{\rm sup}|) \equiv R'$

Note that the notation '$\leftarrow$' indicates enclosure of the result. Moreover, for given matrices $P = (p_{ij}), Q = (q_{ij}) \in \mathbb{F}^{n\times n}$, the notation $\max(P, Q)$ indicates $\max(p_{ij}, q_{ij})$ for all (i, j) pairs, i.e., the maximum is taken componentwise. Here, five matrix multiplications are required for calculating G' until Step 4, and an additional five matrix multiplications for the remaining calculations. Thus, in total, 10 matrix multiplications are required for calculating G' and R'. Therefore, calculating G' and R' involves $20n^3 + O(n^2)$ floating-point operations if the symmetry of G is not taken into account. We compute the upper bounds of $\|R\|_\infty$ and $\|G\|_\infty$ as

$$\|R\|_\infty \le \|R'\|_\infty \le fl_\triangle(\|R'\|_\infty) \equiv \alpha_1, \quad \|G\|_\infty \le \|G'\|_\infty \le fl_\triangle(\|G'\|_\infty) \equiv \alpha_2.$$

If $\alpha_2 \ge 1$, then the verification failed. Hence, we check $\alpha_2 < 1$ or $\alpha_2 \ge 1$ after Step 4. If $\alpha_2 \ge 1$, then the computation prematurely finishes without proceeding to Step 5. Otherwise, we proceed until Step 9 and obtain upper bound r' of r in (13) by

$$r \le fl_\triangle\left(R'e + \frac{\alpha_1}{fl_\triangledown(1-\alpha_2)}G'e\right) \equiv r'. \tag{15}$$

The routine pdsygvx in ScaLAPACK produces computed eigenvalues $\hat{\lambda}_i$ with $\hat{\lambda}_1 \le \hat{\lambda}_2 \le \cdots \le \hat{\lambda}_n$. Therefore, if $\hat{\lambda}_{i+1} - \hat{\lambda}_i > r'_i + r'_{i+1}$ are satisfied for all $i = 1, 2, \ldots, n-1$, then we can separate all of the eigenvalues and determine the order of the eigenvalues correctly.

The test code was developed in the C language with the parallel libraries PBLAS and ScaLAPACK. The solver procedure uses a GEP solver routine (pdsygvx) in ScaLAPACK, whereas the verifier routine uses the matrix multiplication routine (pdgemm) in PBLAS.

Note that the verifier procedure is based primarily on matrix multiplication, whereas the solver procedure consists of complicated procedures, such as Cholesky decomposition, and tridiagonalization. Therefore, the verifier procedure is expected to be moderate in terms of computational time and to be efficient in terms of parallelism, as compared to the solver procedure.

4. Numerical example

4.1. Problem

Numerical examples are presented in this section. All matrix eigenvalue problems stem from the electronic-state calculation software ELSES [27, 28, 29], and the matrix data files appear in the ELSES matrix library [30, 10]. Details are explained in Appendix A. The problems calculated in this section are PPE354, PPE3594, PPE7194, PPE17994, PPE107994, VCNT22500, VCNT225000, and NCCS430080 in the ELSES matrix library. The matrices are those of systems having disordered atomic structures. Disordered systems are important for industrial applications because most industrial materials are disordered, unlike ideal crystal or periodic structures. Consequently, eigenvalues are not degenerate in all of the problems. The number in the problem name indicates the matrix dimension n. For example, the system PPE354 contains $n \times n$ matrices A and B with $n = 354$. All of the matrices A and B in these systems are real symmetric. The systems with the letters 'PPE' are systems of organic polymers of poly-(phenylene-ethynylene) (PPE). The left-hand panel of Fig. 5(a) shows the structural formula of PPE, and the right-hand panel of Fig. 5(b) shows a part of the polymer in a disordered structure. The difference of the matrix size stems from the length of the polymer chain. The system of PPE354 is, for example, a polymer with $N_m = 10$ monomers and $N_{\rm atom} = 12N_m = 120$

atoms. The system VCNT225000 is the system of vibrating carbon nanotube (VCNT). The system NCCS430080 is the system of nano-composite carbon solid (NCCS) [31] and will be explained in the last paragraph of this section.

The characteristic of the eigenvalue distribution can be captured by the following two quantities. One is the difference of sequential approximate eigenvalues $\hat{\delta}_k \equiv \hat{\lambda}_{k+1} - \hat{\lambda}_k$, $k = 1, 2, ..., n-1$, and the other is the eigenvalue count $I(\lambda)$, which is defined on the eigenvalue axis λ as

$$I(\lambda) \equiv \sum_{k=1}^{n} \theta(\lambda_k - \lambda) \tag{16}$$

with the step function

$$\theta(\lambda) \equiv \begin{cases} 1 & (\lambda \geq 0) \\ 0 & (\lambda < 0). \end{cases} \tag{17}$$

In other words, the eigenvalue count $I(\lambda)$ is the number of the eigenvalues that are smaller than λ.

Here, we demonstrate the similarity and the size dependence of the eigenvalue distribution among the organic polymer systems. The organic polymers of PPE354, PPE17994, and PPE107994 are selected. Figures 5(b) and 5(c) show the normalized eigenvalue distribution $I(\lambda)/n$ among these three systems. The three polymers exhibit quite similar curves in Figs. 5(b) and 5(c), and, therefore, the difference $\hat{\delta}_k$ is nearly proportional to $1/n$ ($\hat{\delta}_k \propto 1/n$), as explained in Section 1.

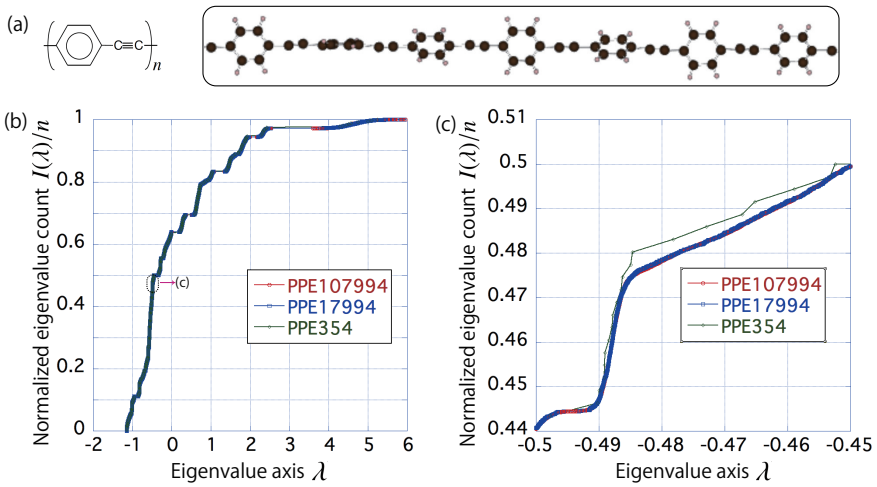

Figure 5: (a) Structural formula (left) and a part of the atomic structure (right) of poly-(phenylene-ethynylene) (PPE). (b) Similarity of eigenvalue distribution in PPE354 (circle), PPE17994 (square), and PPE107994 (diamond). The normalized eigenvalue counts $I(\lambda)/n$ are plotted on the eigenvalue axis λ. (c) Close-up of the local area indicated by the dotted line in (b).

4.2. Numerical results

Tables 1 and 2 show the calculation results on the K computer. First, we focus on the numerical results for the approximate eigenvalues $\hat{\lambda}$ and its upper bound r'. The routine pdsygvx in ScaLAPACK produces $\hat{\lambda}_i$, $i = 1, 2, \ldots, n$ with $\hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \cdots \leq \hat{\lambda}_n$. The vector r' is obtained by (15). Here, we define the radius sum $\rho_k \equiv r'_{k+1} + r'_k$ for $k = 1, 2, \ldots, n-1$. We find m such that $\hat{\delta}_m - \rho_m = \min_{1 \leq k \leq n-1} (\hat{\delta}_k - \rho_k)$. The items "Difference" and "Radius" in Table 1 show $\hat{\delta}_m$ and ρ_m, respectively. As shown in the table, $\hat{\delta}_m > \rho_m$ is satisfied in all of the problems, or all of the disks of $|\lambda_k - \hat{\lambda}_k| < r'_k$ are separated as in Fig. 4. Thus, we can determine the order of eigenvalues in each problem. If $\hat{\delta}_k < \rho_k$ is satisfied for some k, then the two disks of $|\lambda_k - \hat{\lambda}_k| < r'_k$ and $|\lambda_{k+1} - \hat{\lambda}_{k+1}| < r'_{k+1}$ are overlapped and the two exact eigenvalues of λ_k and λ_{k+1} may degenerate.

Figure 6(a) shows the eigenvalue difference $\{\hat{\delta}_k\}$ and the radius sum $\{\rho_k\}$ as a function of the eigenvalue $\{\hat{\lambda}_k\}$ in the case of PPE107994. The radius sum satisfies $\rho_k \leq 10^{-10}$ and is smaller than the difference ($\rho_k < \hat{\delta}_k$). We found the minimality $m = 49,201$ and $\hat{\lambda}_{49201} \approx -0.488$, $\hat{\delta}_{49201} \approx 6.42 \times 10^{-11}$, and $\rho_{49201} \approx 9.17 \times 10^{-12}$. Figure 6(b) shows a close-up of Fig. 6(a) and contains the eigenvalue $\hat{\lambda}_{49201} \approx -0.488$. It is reasonable that the eigenvalue $\hat{\lambda}_{49201}$ appears in the region of $-0.490 < \lambda < -0.485$, because many eigenvalues are densely clustered, and the eigenvalue count $I(\lambda)$ increases rapidly in the region, as shown in Fig. 5(c). The same analysis was also carried out in the case of NCCS430080, which is the largest problem among the present calculations, and the results are shown in Figs. 6(c) and 6(d). The radius sum is smaller than the difference ($\rho_k < \hat{\delta}_k$).

Table 1: Numerical example

Problem name	Matrix dimension (n)	Difference ($\hat{\delta}_m$)	Radius sum (ρ_m)
PPE354	354	6.61×10^{-5}	4.90×10^{-13}
PPE3594	3,594	1.03×10^{-7}	1.33×10^{-12}
PPE7194	7,194	5.55×10^{-8}	1.18×10^{-12}
PPE17994	17,994	5.32×10^{-11}	2.56×10^{-12}
PPE107994	107,994	6.42×10^{-11}	9.17×10^{-12}
VCNT22500	22,500	2.59×10^{-7}	3.20×10^{-10}
VCNT225000	225,000	1.97×10^{-9}	1.64×10^{-9}
NCCS430080	430,080	5.10×10^{-9}	1.61×10^{-9}

Table 2 shows the computational times. The item T_{sol} in Table 2 shows the computing time for pdsygvx in ScaLAPACK. The item T_{veri} shows the computing time for the verification process, mainly, the time for matrix multiplications. Here, the verifier consumes a moderate cost ($T_{\mathrm{veri}} \leq T_{\mathrm{sol}}$), as expected in Section 3.3. More intensive benchmarks, including weak scaling, will be carried out in the future.

Table 2: Elapsed times among the problems. The number of used processor nodes P and the elapsed times for solver T_{sol} and verifier T_{veri} are shown.

Problem name	P	T_{sol}	T_{veri}
PPE354	4	0.32	0.12
PPE3594	4	20.74	4.73
PPE7194	4	118.84	31.74
PPE17994	16	217.91	105.75
PPE107994	600	1009.85	682.92
VCNT22500	64	105.75	59.06
VCNT225000	2025	2625.76	1775.09
NCCS430080	6400	8960.03	3496.56

In conclusion, the verification procedure delivers the intervals that contain the exact eigenvalues ($|\lambda_k - \hat{\lambda}_k| < r'_k$) with the approximate eigenvalues $\hat{\lambda}_k$ and the radius r'_k. We plan to upload the radius data files in ELSES matrix library, as well as the input matrix data and the approximate eigenvalue data. Then, a graph similar to Fig. 6 can be drawn in order to measure the accuracy of the computed solutions.

Finally, the present numerical results are discussed in the context of computational physics. The matrix problem of NCCS430080, the largest matrix problem in the present paper, appears in a previous paper on a nano-composite carbon solid [31]. In general, carbon can form diamond and graphite crystals. The material is composed of graphite-like and diamond-like domains. Figure 7 shows an example of the electronic wavefunction (the highest occupied electronic wavefunction or the wavefunction of the electron that has the highest energy). The atomic structure of Fig. 7 is that of Fig. 2(a) of Ref. [31]. (See Ref. [31] for details.) In the present context, Fig. 7 indicates that the wavefunction is an intermediate wavefunction, as explained in Section 2.1, and lies in the boundary region between graphite-like and diamond-like domains. The a posteriori verification procedure confirms that all of the eigenvalues

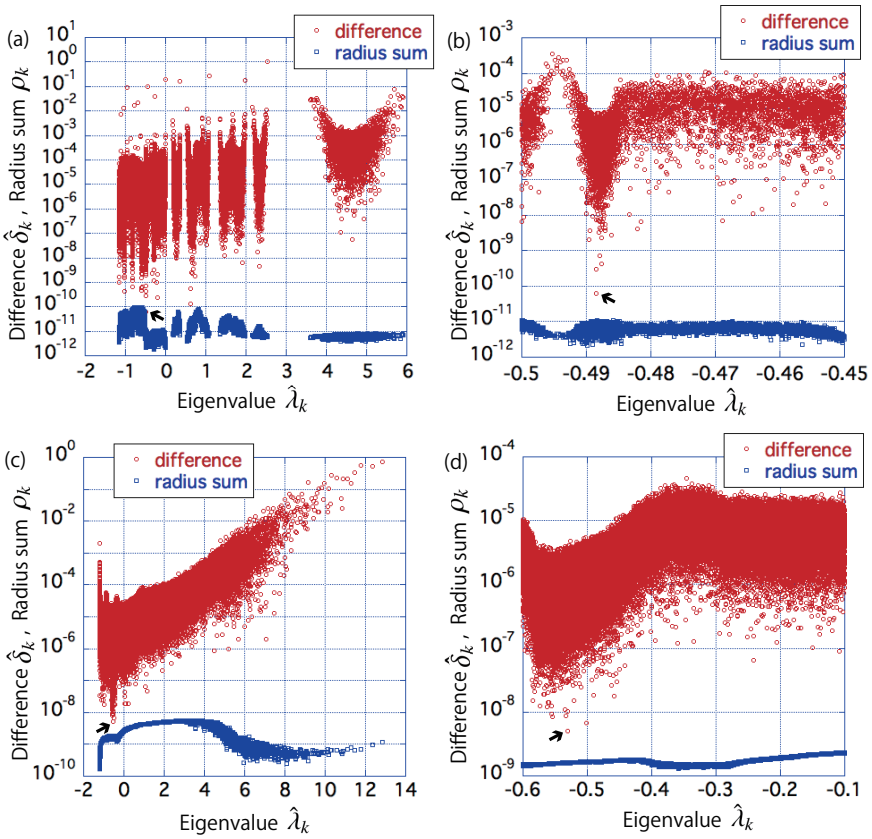

Figure 6: (a) Plot of the eigenvalue difference $\{\hat{\delta}_k\}$ and the radius sum $\{\rho_k\}$ as a function of the eigenvalue $\{\hat{\lambda}_k\}$ in the case of PPE107994. The arrow indicates $\hat{\delta}_m$. (b) Close-up of (a). The arrow indicates $\hat{\delta}_m$. (c) Plot of the eigenvalue difference $\{\hat{\delta}_k\}$ and the radius sum $\{\rho_k\}$ as a function of the eigenvalue $\{\hat{\lambda}_k\}$ in the case of NCCS430080. The arrow indicates $\hat{\delta}_m$. (d) Close-up of (c). The arrow indicates $\hat{\delta}_m$.

are distinguished numerically, and the above physical discussion regarding each wavefunction is meaningful.

5. Summary and overview

The present paper proposes an a posteriori verification method for the generalized eigenvalue problems that appear in large-scale electronic state calculations. The verification procedure gives a rigorous mathematical foundation of numerical reliability. In particular, the present result guarantees that all of the approximate eigenvalues $\{\hat{\lambda}_k\}_k$ are well separated and that the participation ratio value $\{P(\hat{x}_k)\}_k$ and any physical quantity defined for each eigenvector are meaningful. Since the verification procedure consists of simple matrix multiplications, the computational cost is moderate, as compared with that of the solver procedure. Therefore, application researchers can use the verification function with only a moderate increase of the computational cost. Test calculations were carried out on the K computer for real problems with a matrix size of up to $n \approx 4 \times 10^5$.

The next stage of research is the integration of the present verifier routine and solver routines in EigenKernel, in which we can use various solver routines among ScaLAPACK and newer libraries and can compare their approximate solutions in the verification procedure.

Future issues are realizing (i) the verification of eigenvectors and (ii) the refinement of approximate eigenpairs. The refinement procedure will be crucial, in particular, when lower-precision arithmetic, such as half-precision or single-precision arithmetic, is used for calculating an approximate solution as an initial guess. For example, refinement algorithms for the symmetric eigenvalue problem have recently been proposed in [32, 33], which are based on matrix multiplications. Such refinement algorithms enhance application researchers to use lower-precision arithmetic with

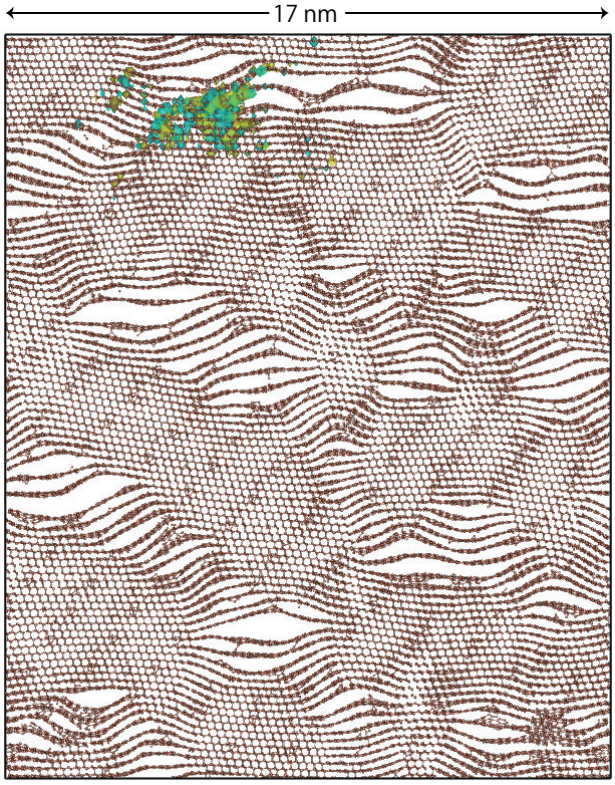

Figure 7: Example of an electronic wavefunction of a nano-composite carbon solid [31]. The wavefunction $\phi(\boldsymbol{r})$ is drawn by the two isosurfaces painted green and yellow. The isovalues are $\phi(\boldsymbol{r}) = \pm C$ ($C > 0$). The wavefunction is 'intermediate', because it exhibits intermediate properties between extended and localized wavefunctions.

satisfactory reliability of the computed results, which will be of great importance in next-generation architecture that is optimized for lower-precision arithmetic.

Acknowledgement

The authors wish to thank the anonymous referees for their valuable comments, which helped to improve our paper significantly. The present study was supported in part by MEXT as Exploratory Issue 1-2 of the Post-K (Fugaku) computer project "Development of verified numerical computations and super high-performance computing environment for extreme researches" using computational resources of the K computer provided by the RIKEN R-CCS through the HPCI System Research project (Project ID: hp180222) and Priority Issue 7 of the Post-K computer project and by JSPS KAKENHI Grant Numbers 16KT0016, 17H02828, and 19H04125.

Appendix A. Generalized eigenvalue problems in electronic state calculations

This section introduces a generalized eigenvalue problem as a numerical foundation of large-scale electronic state calculations. Details can be found in textbooks, such as Ref. [34]. The fundamental Schrödinger-type equation, which is a linear partial differential equation, is written for an electronic wave function $\phi(\boldsymbol{r})$ in real space for a position vector $\boldsymbol{r} = (x, y, z)$ as

$$H\phi(\boldsymbol{r}) = \lambda\phi(\boldsymbol{r}) \tag{A.1}$$

with the Hamilton operator

$$H \equiv -\frac{\hbar^2}{2m}\Delta + V_{\rm eff}(\boldsymbol{r}). \tag{A.2}$$

Here, $\Delta = \partial_x^2 + \partial_y^2 + \partial_z^2$ is the Laplacian, m is the mass of the electron, $\hbar$ ($\approx 1.05^{-34}$ Js) is the Planck constant, and $V_{\rm eff}(\boldsymbol{r})$ is the effective potential, which is a scalar function. The normalization condition

$$\int |\phi(\boldsymbol{r})|^2 = 1 \tag{A.3}$$

is imposed and stems from the fact that the sum of the weight distribution of one electron should be unity.

An eigenvalue λ that indicates the energy of an electron in the material is called an eigenenergy. The k-th eigenpair $(\lambda_k, \phi_k(\boldsymbol{r}))$ is defined for $k = 1, 2, .., n$ in the order of $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Now, we consider as a typical case that $\phi(\boldsymbol{r})$ is expressed as a linear combination of given basic functions

$$\phi(\boldsymbol{r}) = \sum_j^n c_j \chi_j(\boldsymbol{r}), \tag{A.4}$$

where the basis functions $\{\chi_j(\boldsymbol{r})\}$ are normalized to be

$$\int \chi_j^*(\boldsymbol{r})\chi_j(\boldsymbol{r})d\boldsymbol{r} = 1. \tag{A.5}$$

A typical basis function is called atomic orbital and is localized near the position of an atomic nucleus. Since each basis function belongs to one atom, the basis index i is equivalent to the composite indices of an atom index I and an orbital index α ($i \equiv (\mathit{I}, \alpha)$). The orbital index α distinguishes the basis functions that belong to the same atom but differ in shape. Usually, the number of basis functions n is nearly proportional to that of atoms $n_{\rm atom}$ ($n \propto n_{\rm atom}$).

When (A.4) is used for (A.1), the generalized eigenvalue problem (1) appears with the real-symmetric $n \times n$ matrices A and B, where

$$A_{ij} \equiv \int \chi_i^*(\boldsymbol{r})H\chi_j(\boldsymbol{r})d\boldsymbol{r} \tag{A.6}$$

$$B_{ij} \equiv \int \chi_i^*(\boldsymbol{r})\chi_j(\boldsymbol{r})d\boldsymbol{r}. \tag{A.7}$$

The matrix B is positive definite and satisfies $B_{jj} = 1$ and $|B_{ij}| < 1 (i \neq j)$. Hereafter, we consider that the basis functions are real and the matrices A and B are real symmetric. The eigenvectors x_k are real, and the normalization condition of (A.3) is reduced to

$$x_k^{\rm T} B x_k = 1, \tag{A.8}$$

which is called B-normalization.

Here, the simplest theory of the hydrogen molecule (H_2) is demonstrated, as in many textbooks, such as Ref. [35]. The atomic nuclei of the first and second hydrogen atoms are located at $\boldsymbol{r} = \boldsymbol{R}_1$ and $\boldsymbol{R}_2$, respectively. We consider a given localized function $f(\boldsymbol{r})$ with localization center located at $\boldsymbol{r} = 0$. Two basis functions $\chi_1(\boldsymbol{r})$ and $\chi_2(\boldsymbol{r})$ are given as

$$\chi_1(\boldsymbol{r}) \equiv f(\boldsymbol{r} - \boldsymbol{R}_1), \quad \chi_2(\boldsymbol{r}) \equiv f(\boldsymbol{r} - \boldsymbol{R}_2). \tag{A.9}$$

The generalized eigenvalue problem of (1) appears with the 2×2 real symmetric matrices of

$$A \equiv \begin{pmatrix} a & -t \\ -t & a \end{pmatrix}, \quad B \equiv \begin{pmatrix} 1 & s \\ s & 1 \end{pmatrix}. \tag{A.10}$$

Now, we consider a typical case in which a, t, s are positive real numbers and $s < 1$. The off-diagonal element of t or s is the function of the interatomic distance $d \equiv |\boldsymbol{R}_1 - \boldsymbol{R}_2|$ ($t = t(d), s = s(d)$). The eigenvalues are

$$\lambda_1 \equiv \frac{a-t}{1+s}, \quad \lambda_2 \equiv \frac{a+t}{1-s} \tag{A.11}$$

and the eigenvectors are

$$x_1 = \frac{1}{\sqrt{2(1+s)}}\begin{pmatrix}1\\1\end{pmatrix}, \quad x_2 = \frac{1}{\sqrt{2(1-s)}}\begin{pmatrix}1\\-1\end{pmatrix}. \tag{A.12}$$

The matrix B has the eigenvalues $1 \pm s$ and will be not positive definite in the limiting situation of $s \to 1$. One may suspect that the limiting situation can appear, when the distance between the nuclei of atoms is approximately zero ($d \to 0$) and that the two basis functions will be identical ($\chi_1(\boldsymbol{r}) - \chi_2(\boldsymbol{r}) \to 0$). Among real materials, fortunately, the distance d is so large that the limiting situation does not occur.

In general, the matrix size n in generalized eigenvalue problem (1) is nearly proportional to that of the atoms ($n \propto n_{\mathrm{atom}}$). For example, a typical model for the benzene molecule (C_6H_6), called the 'sp' model, gives one basis function for each hydrogen atom (H) and four basis functions for each carbon atom (C). The total number of basis functions or the matrix size n is $n = 1 \times 6 + 4 \times 6 = 30$.

Matrix data of A and B for various materials are stored in the ELSES matrix library [30, 10]. The matrix data were generated by the electronic-state calculation software ELSES [27, 28, 29] with first-principles-based modeled (tight-binding) electronic-state theory. The atomic unit is used for the energy among the data files. For example, the matrix problem for a benzene molecule (C_6H_6) in the above model is stored as 'BNZ30'. For many problems, the approximate eigenvalues $\{\hat{\lambda}_k\}$ are uploaded, as well as the matrix data of A and B, for the convenience of the researcher. The sparsity of the stored matrix data of A_{ij} and B_{ij} is explained briefly. As explained above, the indices i and j are the composite indices of the atom indices I and J and the orbital indices α and β, respectively ($i \equiv i(I,\alpha)$, $j \equiv j(J,\beta)$)). Therefore, an element of the matrices A and B is expressed by the four indices as $A_{I\alpha;J\beta}$ and $B_{I\alpha;J\beta}$, respectively. Since a matrix element value decreases quickly and monotonically as a function of the inter-atomic distance between the I-th and J-th atoms (r_{IJ}), a cutoff distance r_{cut} can be introduced. A matrix element, $A_{I\alpha;J\beta}$ or $B_{I\alpha;J\beta}$, is ignored, if $r_{IJ} > r_{\mathrm{cut}}$, which makes the matrices sparse. More information on the data file in the ELSES matrix library is found in Ref. [10].

As a future issue, we should consider the numerical error in the input matrix data of A and B, because this error may affect the final conclusion. The possible numerical error can be decomposed into the two terms

$$A_{\mathrm{exact}} - A = \delta A_{\mathrm{cut}} + \delta A_{\mathrm{cal}} \tag{A.13}$$

$$B_{\mathrm{exact}} - B = \delta B_{\mathrm{cut}} + \delta B_{\mathrm{cal}}, \tag{A.14}$$

where A_{exact} and B_{exact} are the exact (theoretical) matrix data. The error terms δA_{cut} and δB_{cut} are called cutoff errors and stem from the cutoff procedure explained in the previous paragraph. The maximum element of δA_{cut} or δB_{cut} is on the order of 10^{-4} in the case of PPE17994, for example. The cutoff error term will be eliminated when the full matrix data are adopted in the input matrices. The full matrix data are not stored in the ELSES matrix library, because these data consume a large amount of disk space. We intend to perform verification using the full matrix data when we integrate the verifier routines into the simulation software (ELSES), which generates the matrix data and includes the solver routine. The procedure with the full matrix data will not increase the computational cost, because all of the procedures use the dense-matrix algorithms. The error terms δA_{cal} and δB_{cal}, on the other hand, are called calculation errors and stem from the generating procedure of the matrices. The matrix B is defined as the integral in (A.7). The three-dimensional numerical integral of (A.7) is obtained for the Slater-type functions $\{\chi_i(\boldsymbol{r})\}_i$ [36] in prolate spheroidal coordinates, which is reviewed in Section II of Ref. [37]. The matrix A is calculated from the matrix B in a modeled (atom superposition and electron delocalization tight-binding) theory [38, 39, 28]. Evaluating the calculation errors δA_{cal} and δB_{cal} is an interesting topic and will be discussed in the near future.

[1] EigenKernel, `https://github.com/eigenkernel/`.

[2] H. Imachi, T. Hoshi, Hybrid numerical solvers for massively parallel eigenvalue computations and their benchmark with electronic structure calculations, J. Inf. Process 24 (2016) 164–172.

[3] K. Tanaka, H. Imachi, T. Fukumoto, T. Fukaya, Y. Yamamoto, T. Hoshi, EigenKernel - A middleware for parallel generalized eigenvalue solvers to attain high scalability and usability, Japan J. Indust. Appl. Math. 36 (2019) 719–742.
[4] N. Yamamoto, A simple method for error bounds of eigenvalues of symmetric matrices, Linear Algebra Appl. 324 (1–3) (2001) 227–234.
[5] S. Miyajima, T. Ogita, S. M. Rump, S. Oishi, Fast verification for all eigenpairs in symmetric positive definite generalized eigenvalue problem, Reliable Computing 14 (2010) 24–45.
[6] S. Miyajima, Numerical enclosure for each eigenvalue in generalized eigenvalue problem, J. Comput. Appl. Math. 236 (9) (2012) 2545–2552.
[7] T. Yamamoto, Error bounds for approximate solutions of systems of equations, Japan J. Appl. Math. 1 (1) (1984) 157–171.
[8] J. Dongarra, Issue and solutions for extreme scale computing, in: HPC Asia 2018, Tokyo, Japan, 2018.
[9] A. Alvermann, A. Basermann, H.-J. Bungartz, C. Carbogno, D. Ernst, H. Fehske, Y. Futamura, M. Galgon, G. Hager, S. Huber, T. Huckle, Ida, A. Imakura, M. Kawai, S. Köcher, M. Kreutzer, P. Kus, B. Lang, H. Lederer, V. Manin, A. Marek, K. Nakajima, L. Nemec, K. Reuter, M. Rippl, M. Röhrig-Zöllner, T. Sakurai, M. Scheffler, C. Scheurer, F. Shahzad, D. S. Brambila, J. Thies, G. Wellein, Benefits from using mixed precision computations in the ELPA-AEO and ESSEX-II eigensolver projects, Japan J. Indust. Appl. Math. 36 (2019) 699–717.
[10] T. Hoshi, H. Imachi, A. Kuwata, K. Kakuda, T. Fujita, H. Matsui, Numerical aspect of large-scale electronic state calculation for flexible device material, Japan J. Indust. Appl. Math. 36 (2019) 685–698.
[11] R. Bell, P. Dean, Atomic vibrations in vitreous silica, Disc. Faraday Soc. 50 (1970) 55–61.
[12] T. Fujiwara, T. Mitsui, S. Yamamoto, Scaling properties of wave functions and transport coefficients in quasicrystals, Phys. Rev. B 53 (1996) R2910–R2913.
[13] ScaLAPACK, `http://www.netlib.org/scalapack/`.
[14] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, R. C. Whaley, ScaLAPACK Users' Guide, Society for Industrial and Applied Mathematics, 1997.
[15] ELPA(= eigenvalue solvers for petaflop-applications), `https://elpa.mpcdf.mpg.de/`.
[16] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H. Bungartz, H. Lederer, The ELPA library - scalable parallel eigenvalue solutions for electronic structure theory and computational science, J. Phys. Condens. Matter 26 (2014) 213201.
[17] EigenExa, `http://www.r-ccs.riken.jp/labs/lpnctrt/en/projects/eigenexa/`.
[18] T. Imamura, Y. Hirota, T. Fukaya, S. Yamada, M. Machida, Eigenexa: high performance dense eigensolver, present and future, in: 8th International Workshop on Parallel Matrix Algorithms and Applications (PMAA14), Lugano, Switzerland, 2014.
[19] F. Fritz Haber Institute *ab initio* molecular simulations), `https://aimsclub.fhi-berlin.mpg.de/`.
[20] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, *Ab initio* molecular simulations with numeric atom-centered orbitals, Comp. Phys. Comm. 180 (2009) 2175–2196.
[21] S. M. Rump, Fast and parallel interval arithmetic, BIT Numer. Math. 39 (3) (1999) 539–560.
[22] C-XSC – A C++ Class Library for Extended Scientific Computing, ver. 2.5.4, `http://www.xsc.de/`.
[23] S. M. Rump, INTLAB – INTerval LABoratory, in: T. Csendes (Ed.), Developments in Reliable Computing, Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104.
[24] J. H. Wilkinson, Rigorous error bounds for computed eigensystem, Computer J. 4 (1961) 230–241.
[25] D. Lee, T. Hoshi, T. Sogabe, Y. Miyatake, S.-L. Zhang, Solution of the *k*-th eigenvalue problem in large-scale electronic structure calculations, J. Comp. Phys 371 (2018) 618–632.
[26] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.
[27] ELSES(= Extra-Large-Scale Electronic Structure calculation), `http://www.elses.jp/`.
[28] T. Hoshi, S. Yamamoto, T. Fujiwara, T. Sogabe, S.-L. Zhang, An order-*N* electronic structure theory with generalized eigenvalue equations and its application to a ten-million-atom system, J. Phys.: Condens. Matter (2012) 24/165502, 1–5.
[29] T. Hoshi, H. Imachi, K. Kumahata, M. Terai, K. Miyamoto, K. Minami, F. Shoji, Extremely scalable algorithm for 10^8-atom quantum material simulation on the full system of the k computer, Proc. ScalA16 in SC16 (2016) 33–40.
[30] ELSES Matrix Library, `http://www.elses.jp/matrix/`.
[31] T. Hoshi, Y. Akiyama, T. Tanaka, T. Ohno, Ten-million-atom electronic structure calculations on the K computer with a massively parallel order-N theory, J. Phys. Soc. Jpn. 82 (2013) 023710/1–4.
[32] T. Ogita, K. Aishima, Iterative refinement for symmetric eigenvalue decomposition, Japan J. Indust. Appl. Math. 35 (3) (2018) 1007–1035.
[33] T. Ogita, K. Aishima, Iterative refinement for symmetric eigenvalue decomposition II: clustered eigenvalues, Japan J. Indust. Appl. Math. 36 (2) (2019) 435–459.
[34] R. M. Martin, Electronic Structure - Basic Theory and Practical Methods, Cambridge University Press, 2004.
[35] P. Atkins, R. Friedman, Molecular Quantum Mechanics, 5th Ed., Oxford University Press, 2005.
[36] J. C. Slater, Atomic shielding constants, Phys. Rev. 36 (1930) 57–64.
[37] M. Lesiuk, R. Moszynski, Reexamination of the calculation of two-center, two-electron integrals over Slater-type orbitals. I. Coulomb and hybrid integrals, Phys. Rev. E 90 (2014) 063318/1–13.
[38] K. Nath, A. B. Anderson, Atom-superposition and electron-delocalization tight-binding band theory, Phys. Rev. B 41 (1990) 5652–5660.
[39] G. Calzaferri, R. Rytz, The band structure of diamond, J. Phys. Chem. 100 (1996) 11122–11124.